GURLS:
A Least Squares Library for Supervised Learning

(and the roadmap toward the integration within the iCub software development process)

Matteo Santoro (*Laboratory for Computational and Statistical Learning*)

VVV13, Veni Vidi Vici 2013, the iCub Summer School – June 18th 2013 – Sestri Levante

L. Rosasco, A. Tacchetti, P.K. Mallapragada

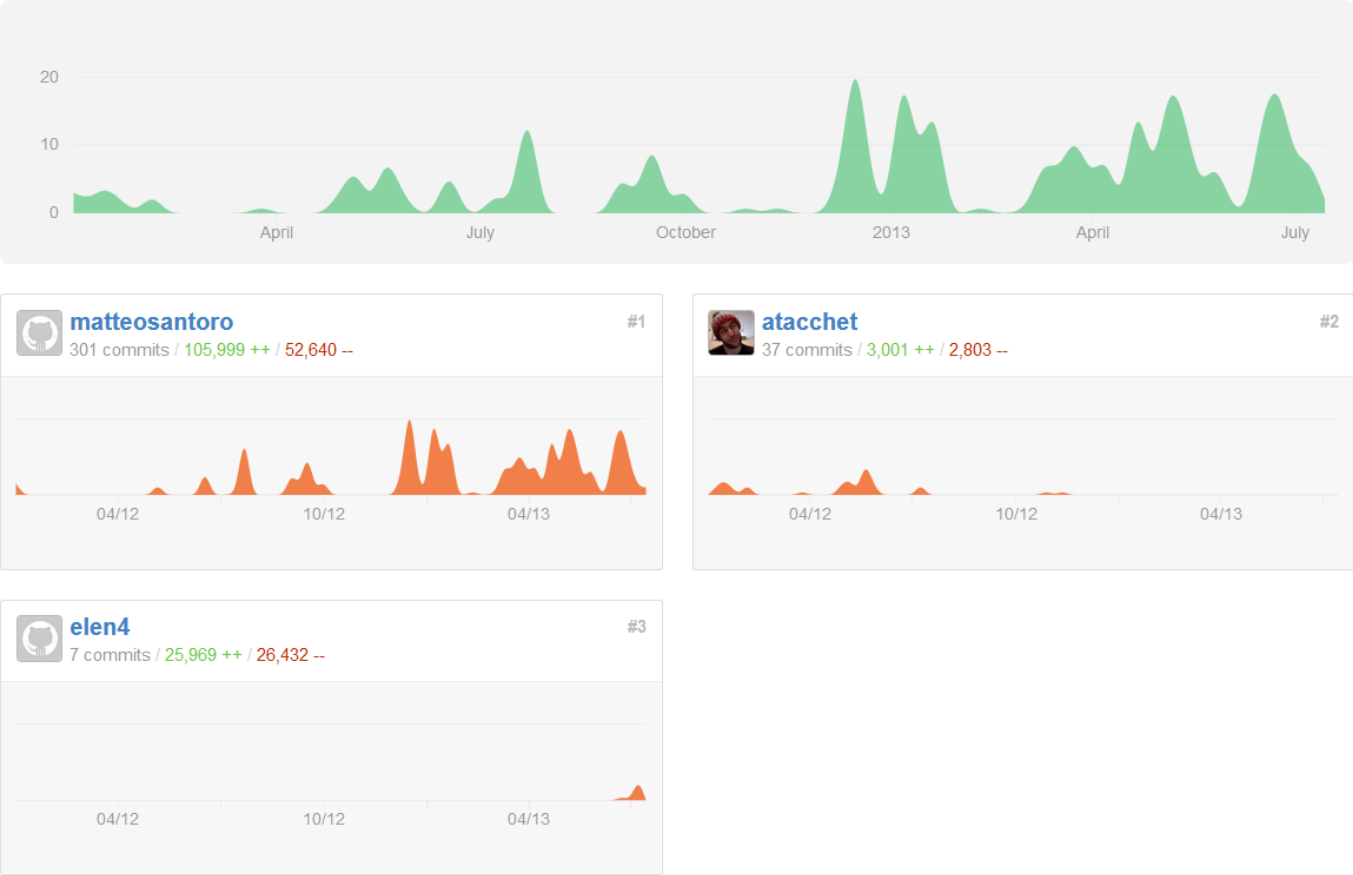
E. Ceseracciu, C. Ciliberto, S. Fanello

Summary

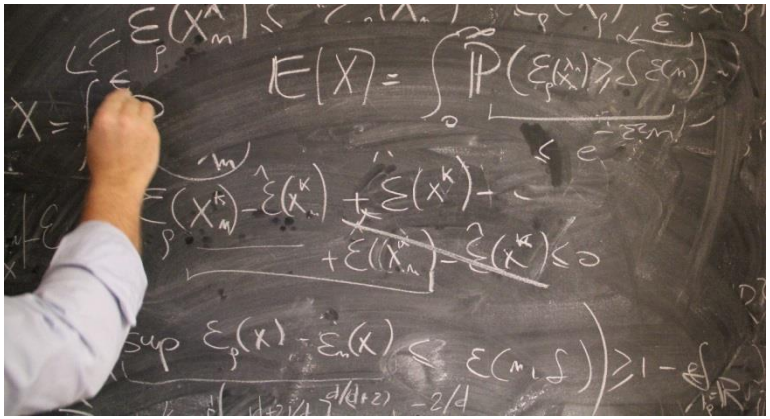
- Basic facts
- Main ideas
- Design and Architecture of the Library
- Examples (*Matlab and C++ code*)
- Experiments and Results

The Library

- https://github.com/LCSL/GURLS



Useful Links and References



- http://lcsl.mit.edu
- http://lcsl.mit.edu/gurls.html

GURLS: a Least Squares Library for Supervised Learning. Andrea Tacchetti, Pavan K Mallapragada, Matteo Santoro, Lorenzo Rosasco. ArXiv:1303.0934

A revised version will appear soon on the Journal of Machine Learning Research

Documentation

- https://github.com/LCSL/GURLS/blob/master/gurls-manual.pdf
- https://github.com/LCSL/GURLS/wiki
- http://cbcl.github.io/GURLS/

Organization of the Library

- **GURLS**
  - a MATLAB software library for regression and (multiclass) classification based on the Regularized Least Squares (RLS) loss function. Datasets that fit into your computer's memory should be handled with this package.
- **bGURLS** (b is for big)
  - a MATLAB software library that allows to use RLS on very large matrices by means of memory-mapped storage and a simple distributed task manager.
- **GURLS++**
  - a C++ standalone implementation of GURLS, with additional simple API's for specific learning pipelines
- **bGURLS++**
  - a C++ standalone implementation of bGURLS.

Initial Requirements

- **Speed**
  - Fast training/testing procedures (online, batch, randomized, distributed) for learning problems with potentially large/huge number of points, features and especially outputs (e.g. classes).
- **Memory**
  - Flexible data management to work with large datasets by means of memory-mapped storage.
- **Performance**
  - State of the art results in high-dimensional multi-output problems (e.g. object recognition tasks with tens or hundreds of classes, where the input have dense features).
- **Usability and modularity**
  - Easy to use and to expand library.

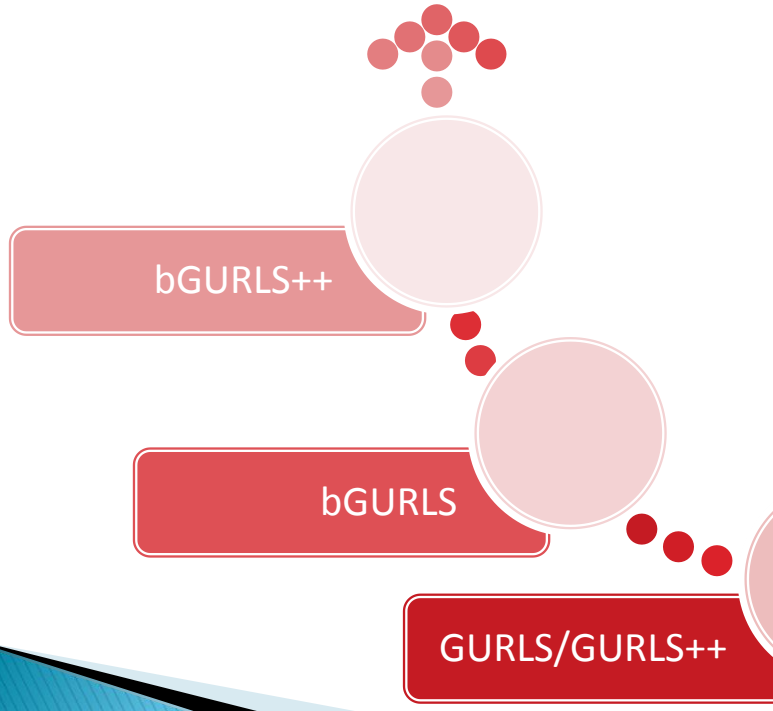
Overview

KDDCUP99
- ✓ 4.900.000 training examples
- ✓ 127 dimensional feature space
- ✓ 2 classes

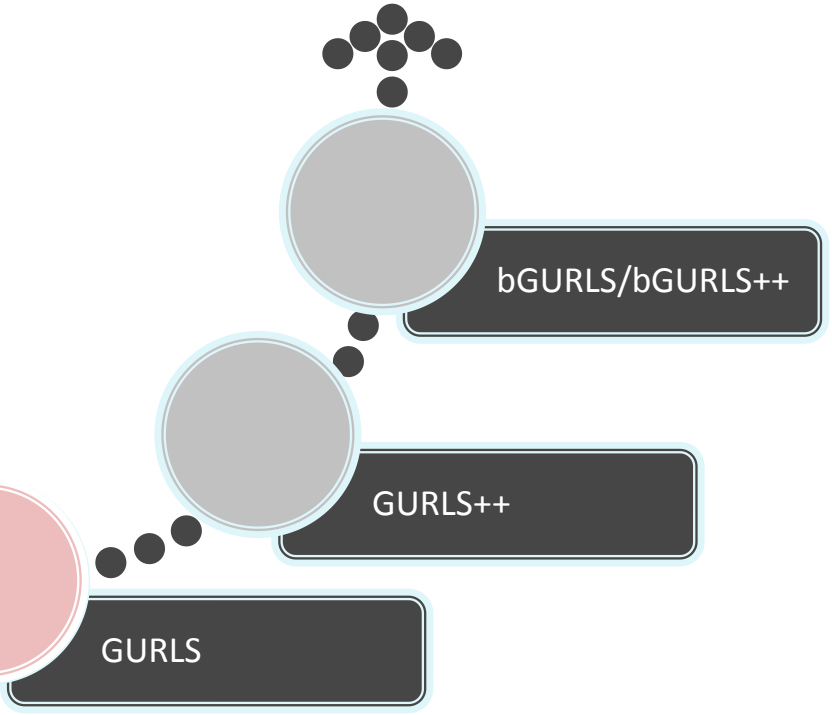
ImageNet
- ✓ 1.000.000 training examples
- ✓ 7.000 dimensional feature space
- ✓ 1.000 classes
- ✓ 300Gb of training data

Big Learning

Towards Parallelism

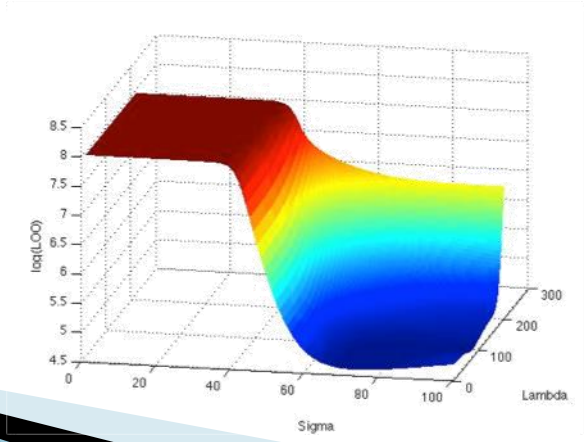


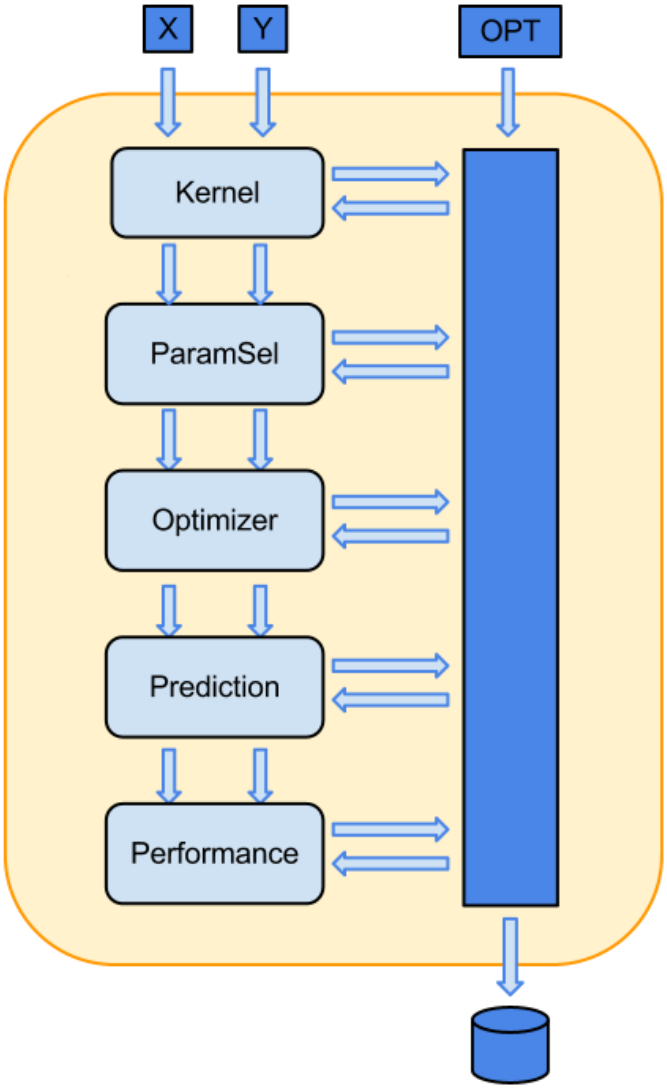
Increasing dataset size



Overview

- Easy experiment management
- New data structure bigarray
- Different kernels
- Different performance measures
- Different training strategies
- Different cross-validation methods
- Completely modular
- Implements and exploits all RLS properties





Available Tasks in GURLS

Task category	Description	Available tasks
split	Splits data into one or more pair of training and validation sets	ho
paramsel	Performs selection of the regularization parameter lambda and, if using Gaussian kernel, also of the kernel parameter sigma	fixlambda, loocvprimal, loocvdual, hoprimal, hodual, siglam, siglamho, bfprimal, bfdual, calibratesgd, hoprimalr, hodualr, horandfeats, gpregrLambdaGrid, gpregrSigLambGrid, loogpregr, hogpregr, siglamhogpregr, siglamloogpregr,
kernel	Builds the symmetric kernel matrix to be used for training	chisquared, linear, load, randfeats, rbf
rls	Computes the optimizer	primal, dual, auto, pegasos, primalr, dualr, randfeats, gpregr
predkernel	Builds the train-test kernel matrix	traintest
pred	Predicts the labels	primal, dual, randfeats, gpregr
perf	Assess prediction performance	macroavg, precrec, rmse
conf	Computes a confidence for the highest scoring class	maxscore, gap, boltzmangap, boltzman

The «pipeline» concept

- A sequence of «tasks»:
  - *{'kernel:linear', 'paramsel:loocvdual', 'rls:dual', 'pred:dual', 'perf:macroavg'}*
- A sequence of processing policies:
  - *{ C1, C2, C3, C4, C5}*

- A processing policy “Ci” may be:
  - 0 = Ignore
  - 1 = Compute
  - 2 = Compute and save
  - 3 = Load from file
  - 4 = Explicitly delete

The «pipeline» concept

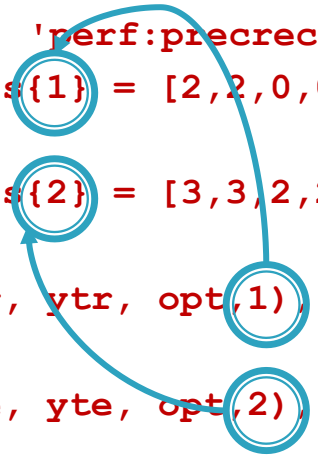
- GURLS includes an «engine» (which is in gurls.m) responsible for the parsing of the pipeline description and execute a whole machine learning experiment.

```
name = 'ExampleExperiment';
opt = defopt(name);
opt.seq = {'paramsel:loocvprimal','rls:primal','pred:primal', ...
           'perf:precrec','perf:macroavg'};
opt.process{1} = [2,2,0,0,0];

opt.process{2} = [3,3,2,2,2];

gurls (Xtr, ytr, opt,1)

gurls (Xte, yte, opt,2)
```



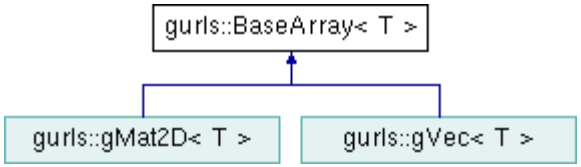
GURLS++

- A complete (and revised!) C++ porting of the original Matlab package.
- (Only) a bit more complicated to install.
- Requires some external dependecies:
  - Boost (mainly for serialization)
  - Blas/Lapack (you may choose the one you prefer)

- See the wiki tutorial How to install the C Libraries

- Good luck (C++ is not Matlab)

Linear Algebra in GURLS++



Basic Linear Algebra Subroutines

Name	Description	Examples
Level-1 BLAS	Vector Operations	$C = \sum X_i Y_i$
Level-2 BLAS	Matrix-Vector Operations	$B_i = \sum_k A_{ik} X_k$
Level-3 BLAS	Matrix-Matrix Operations	$C_{ij} = \sum_k A_{ik} B_{kj}$

$$\begin{bmatrix} L & A & P & A & C & K \\ L & -A & P & -A & C & -K \\ L & A & P & A & -C & -K \\ L & -A & P & -A & -C & K \\ L & A & -P & -A & C & K \\ L & -A & -P & A & C & -K \end{bmatrix}$$

Available Tasks in GURLS++

Task category	Description	Class name	Available subclasses (tasks)
split	Splits data into one or more pair of training and validation sets	Split	Ho
paramsel	Performs selection of the regularization parameter lambda and, if using Gaussian kernel, also of the kernel parameter sigma	ParamSelection	LoocvDual, LoocvPrimal, HoDual, HoPrimal, SiglamHo, Siglam, FixLambda FixSigLam, HoPrimalr, HoDualr, LooGPRegr, HoGPRegr, SigLamLooGPRegr, SigLamHoGPRegr, CalibrateSGD
kernel	Builds the symmetric kernel matrix to be used for training	Kernel	ChisquaredKernel, LinearKernel, RBFKernel
rls	Computes the optimizer	Optimizer	RLSPrimal, RLSDual, RLSAuto, RLSPegasos, RLSPrimalr, RLSDualr, RLSGPRegr, RLSPrimalRecInit, RLSPrimalRecUpdate
predkernel	Builds the train-test kernel matrix	PredKernel	TrainTest
pred	Predicts the labels	Prediction	PredPrimal, PredDual, PredGPRegr
perf	Assess prediction performance	Performance	MacroAvg, PrecisionRecall, Rmse
conf	Computes a confidence for the highest scoring class	Confidence	ConfMaxScor

Recursive Regularized Least Squares

- Recursive RLS is an algorithm for efficient update of the exact estimator when the data are given sequentially as in online learning.
- For any new input-output pair the (regularized) inverse of the kernel matrix in the primal space is updated just via matrix/vector multiplication

Algorithm 1 Recursive RLS

given:

the initial RLS estimator $\mathbf{w}_{n_0}$

the inverse of the (regularized) kernel matrix, $\mathbf{C}_{n_0}^{-1}$

an order sequence of input--output pairs $\{(\mathbf{x}_{n_0+1}, \mathbf{y}_{n_0+1}), \ldots, (\mathbf{x}_{n_{max}}, \mathbf{y}_{n_{max}})\}$

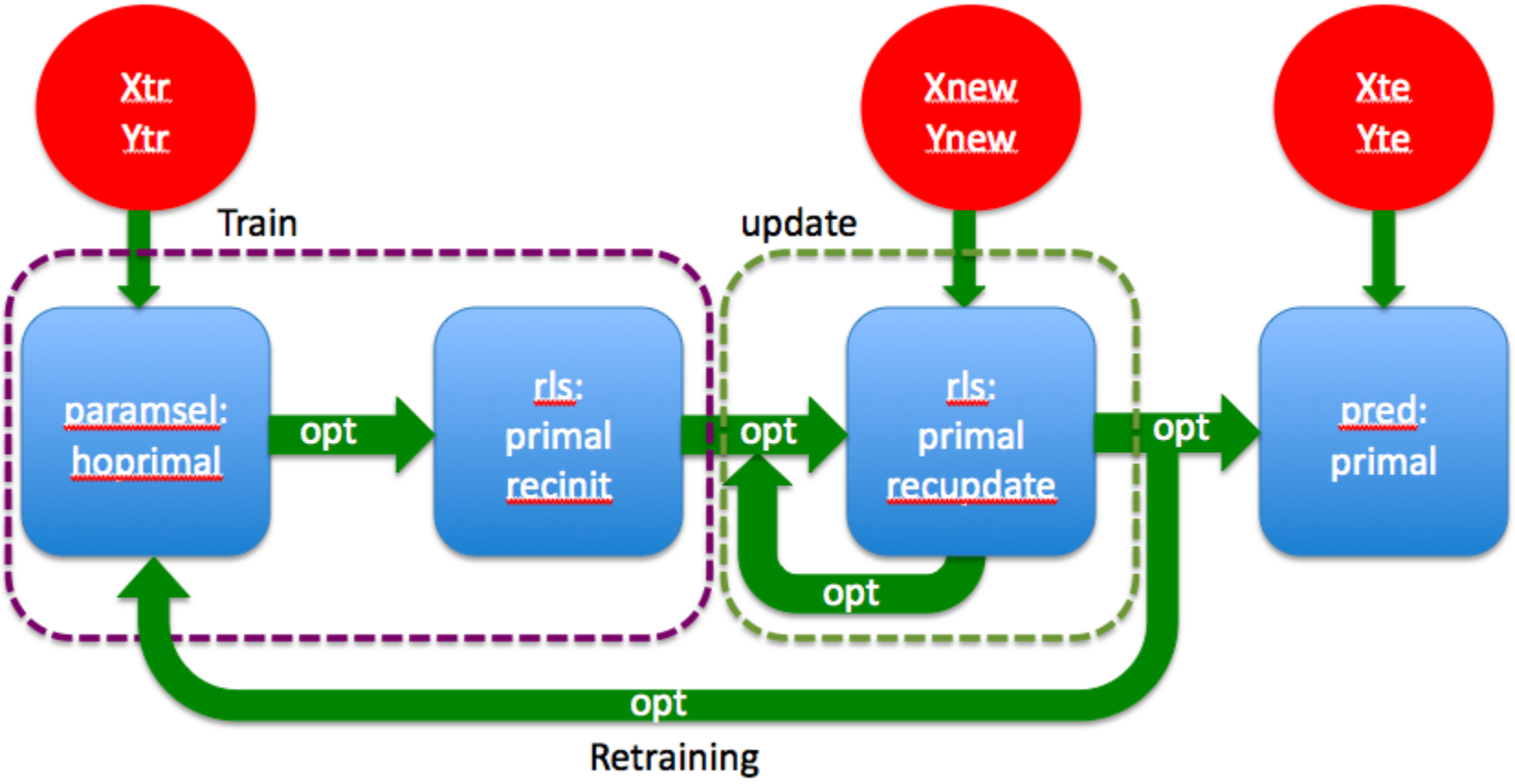
for $\mathbf{n} = \mathbf{n_0}, \ldots, \mathbf{n_{max}} - \mathbf{1}$ **do**

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \frac{\mathbf{C_n^{-1}}}{\mathbf{1} + \mathbf{x_n^T C_n^{-1} x_n}} \mathbf{x_n}(\mathbf{y_n} - \mathbf{x_n^T w_n})$$

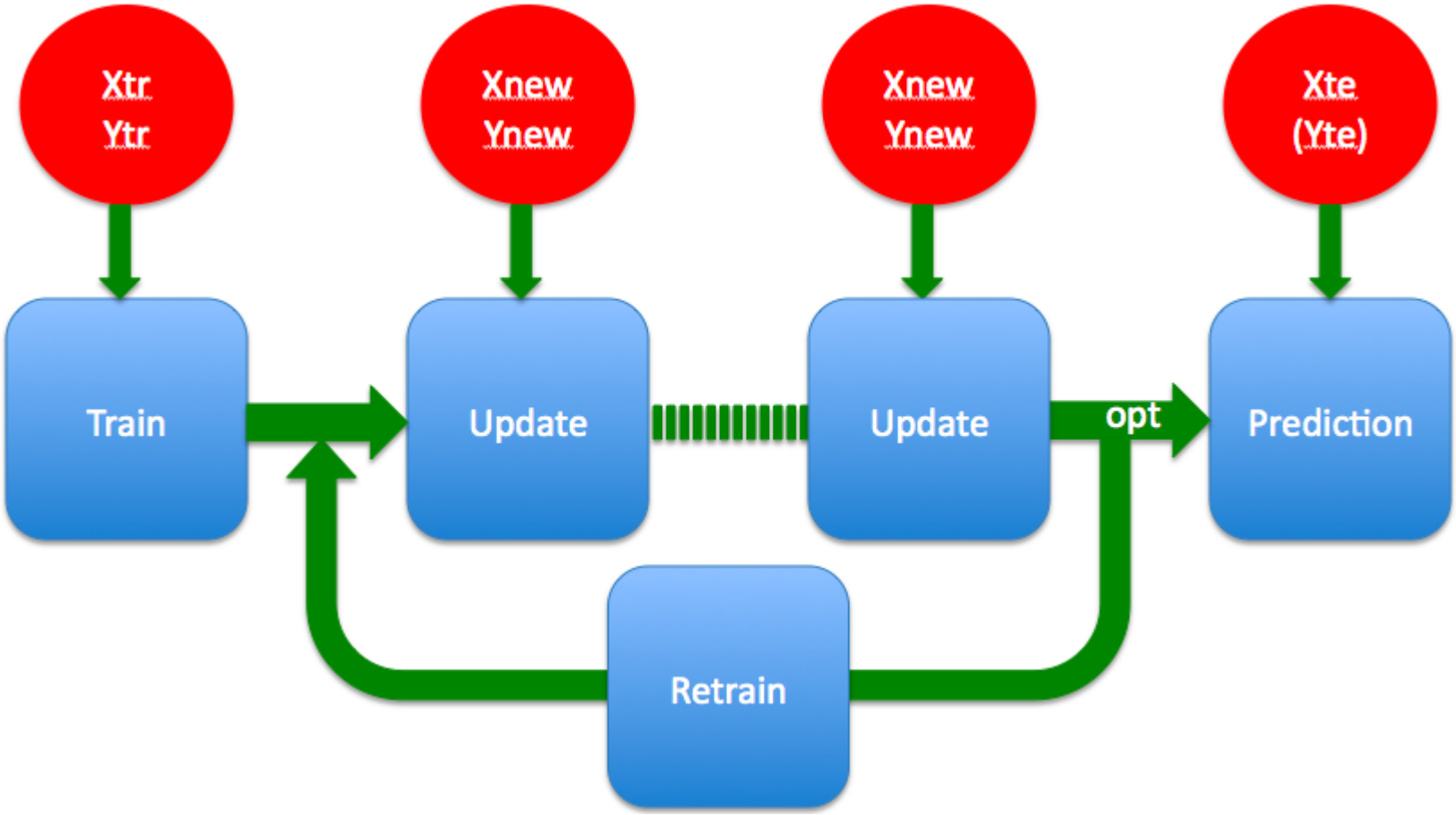
end for

return $\mathbf{w}_{n_{max}}$

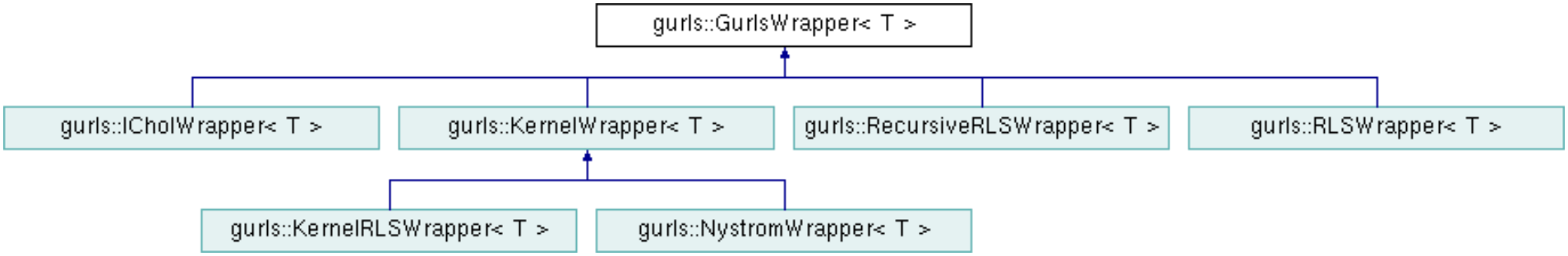
Beyond the pipeline…

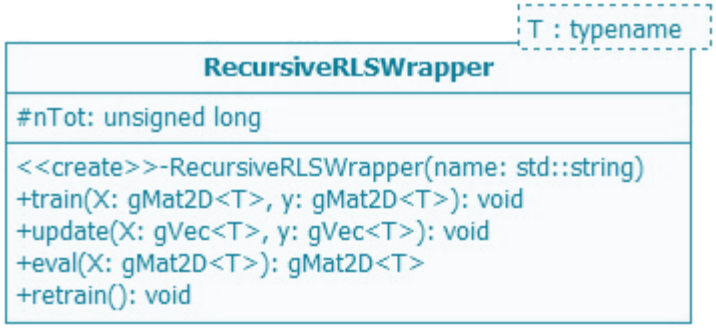


Beyond the pipeline…

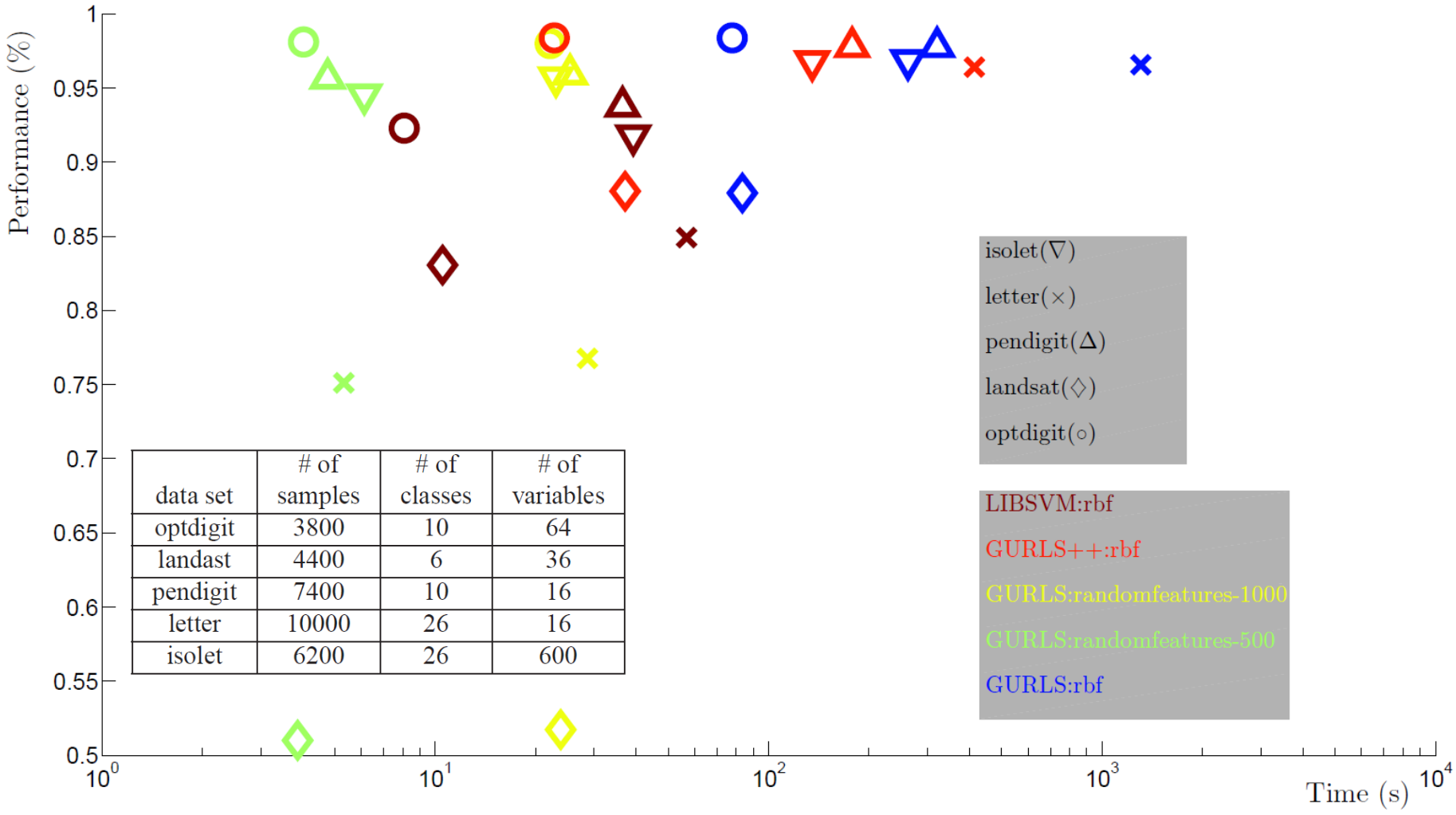


Gurls Wrappers



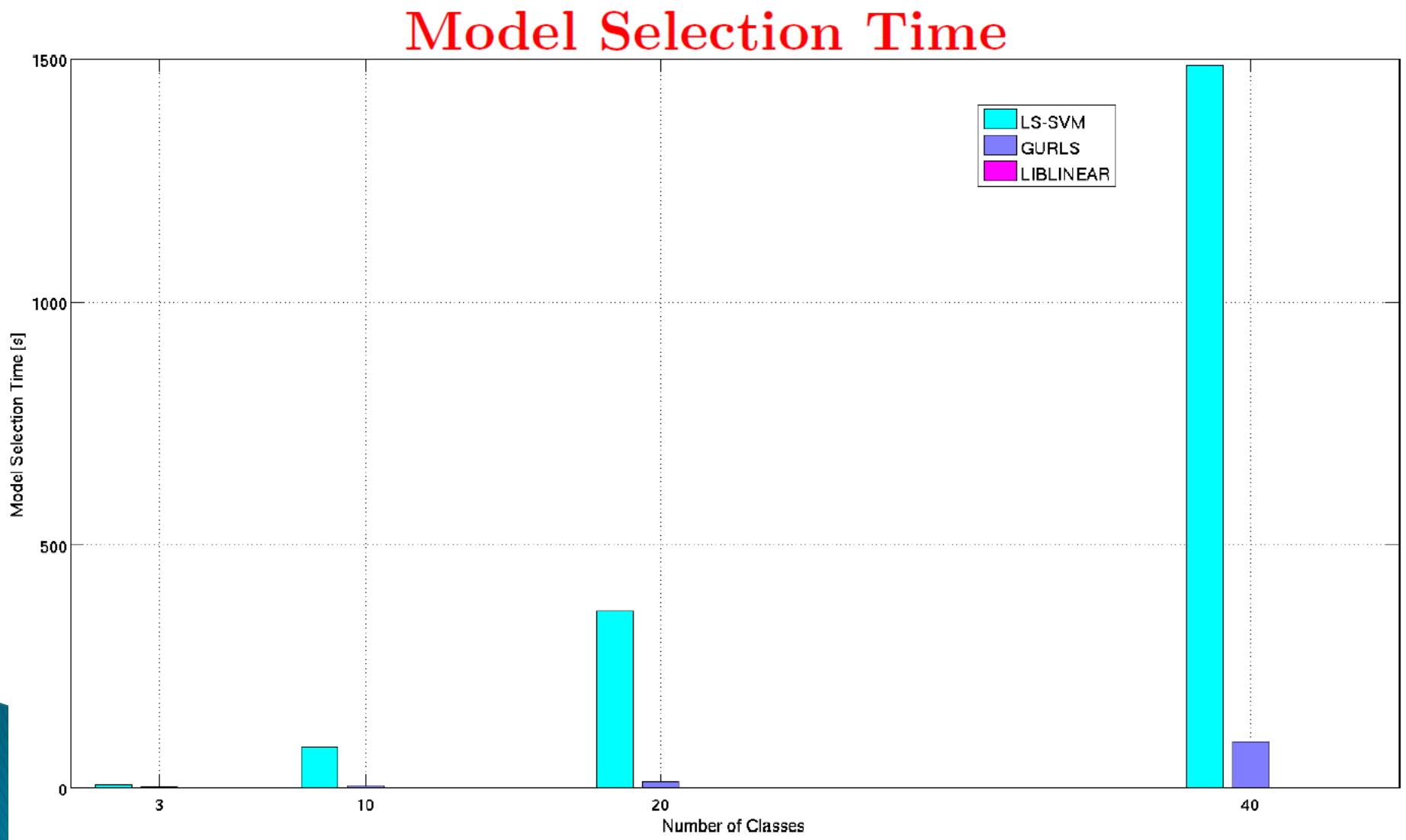


Experimental Results

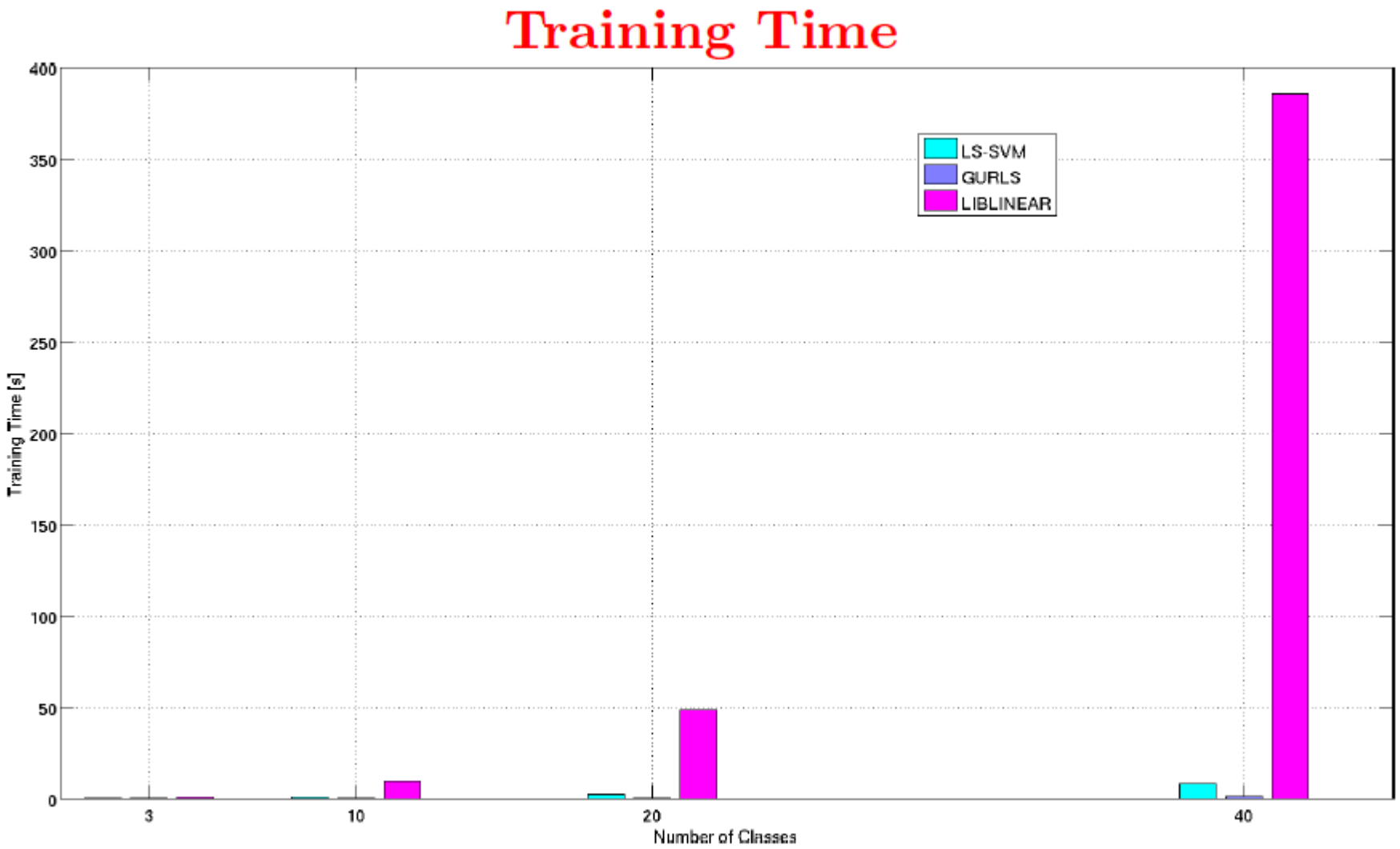


data set	# of samples	# of classes	# of variables
optdigit	3800	10	64
landast	4400	6	36
pendigit	7400	10	16
letter	10000	26	16
isolet	6200	26	600

Experimental Results



Experimental Results



Experimental Results

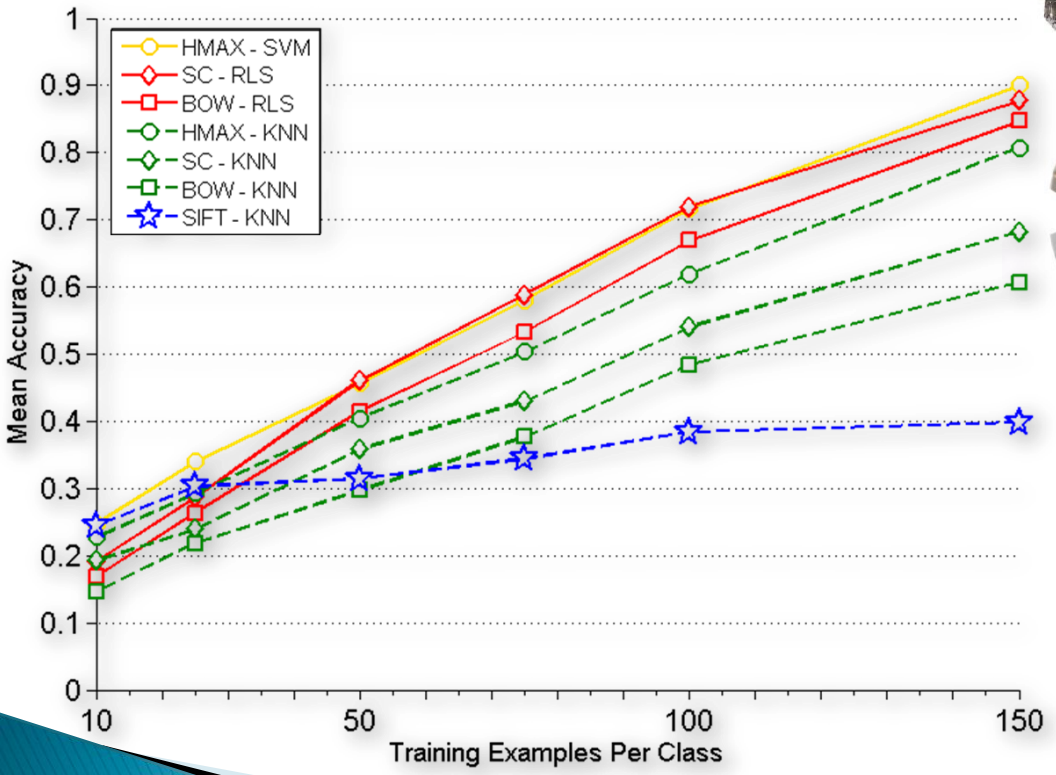
PubFig83

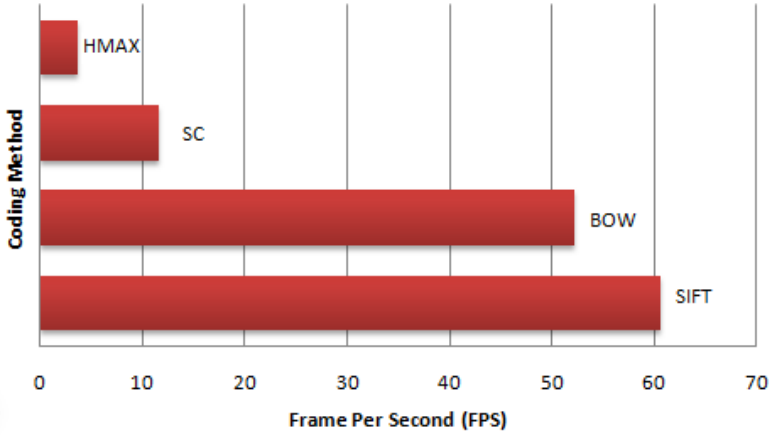
http://www.eecs.harvard.edu/ zak/pubfig83/

Package	*Kernel*	*Accuracy*	*Time*
GURLS	Linear	87%	0h13m
LIBSVM	Linear	76%	5h20m
GURLS	RBF with selection	88%	5h51m
GURLS	RBF = 25th PCT of DST	87%	0h14m
LIBSVM	RBF = 25th PCT of DST	76%	4h18m

A visual surveillance example

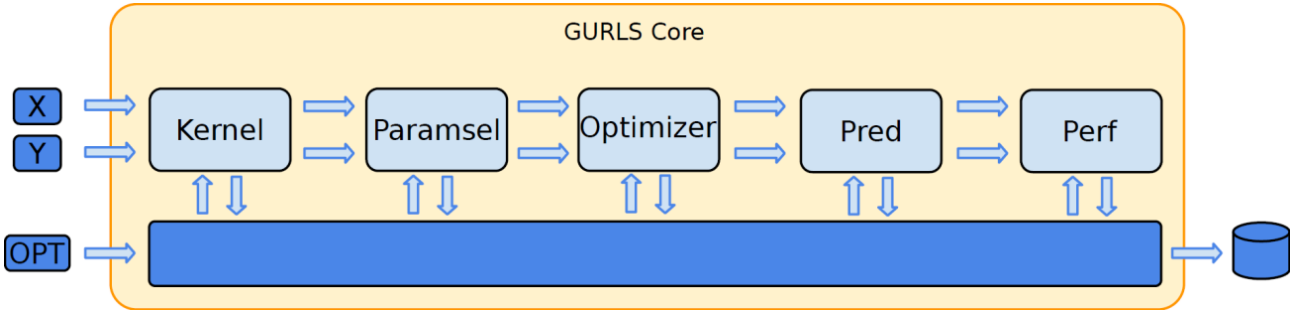
A robotic example

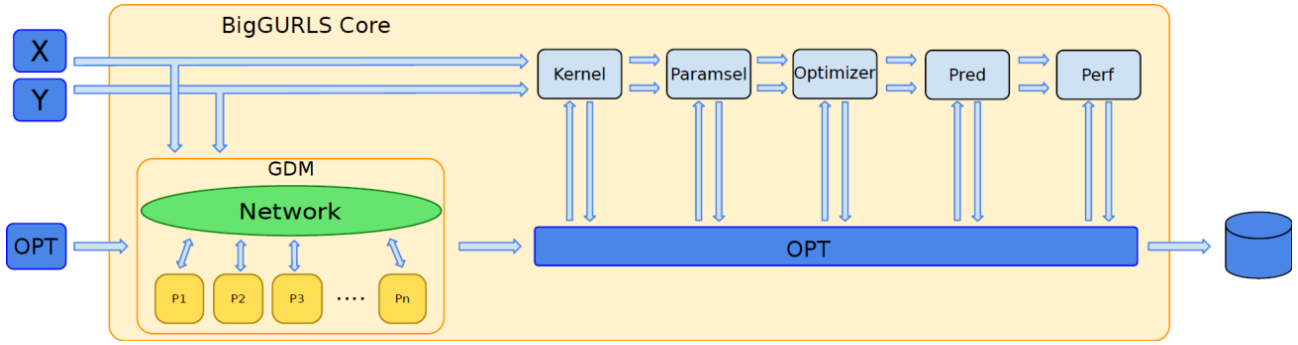




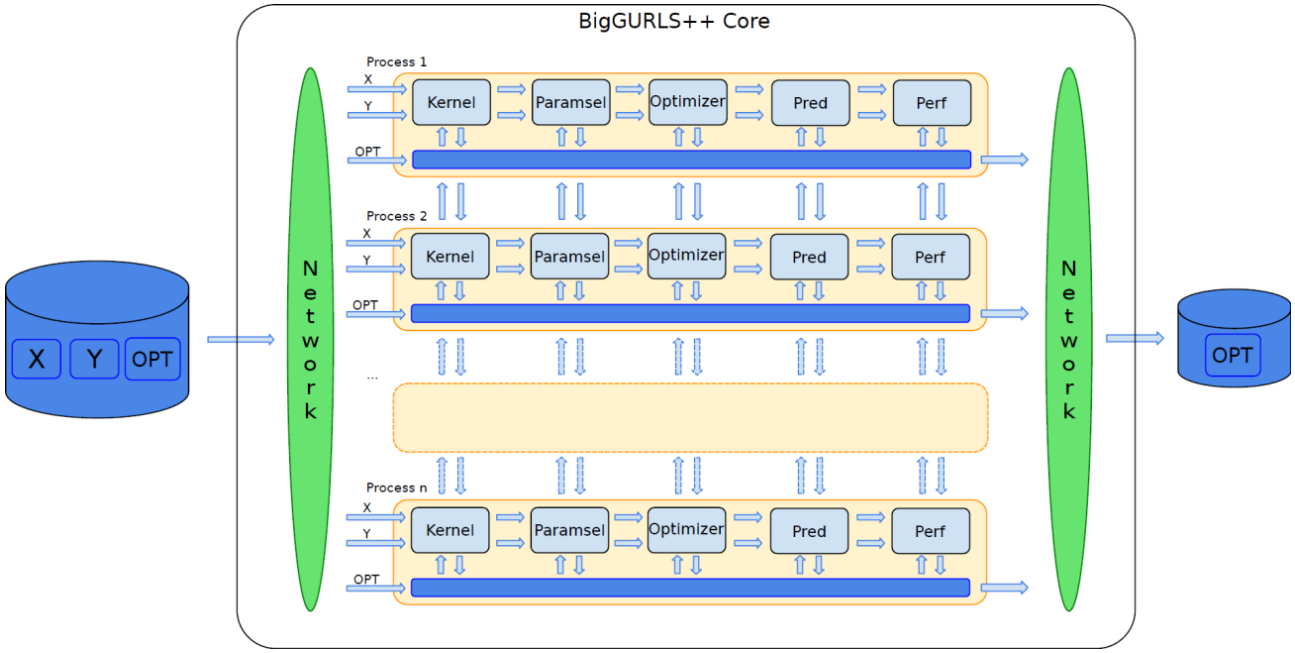
C Ciliberto, S Fanello, M Santoro, L Natale, G Metta and L Rosasco. On the Impact of Learning Hierarchical Representations for Visual Recognition in Robotics. To appear in the Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)

A glimple at bGURLS/bGURLS++





A glimple at bGURLS/bGURLS++



Contacts

- matteo.santoro@gmail.com